Graph Enhanced Memory Networks for Sentiment Analysis

Zhao Xu[1], Romain Vial[1,2], and Kristian Kersting[3]

[1] NEC Laboratories Europe, Germany
zhao.xu@neclab.eu
[2] MINES ParisTech, PSL Research University, France
romain.vial@mines-paristech.fr
[3] Technical University of Darmstadt, Germany
kersting@cs.tu-darmstadt.de

Abstract. Memory networks model information and knowledge as memories that can be manipulated for prediction and reasoning about questions of interest. In many cases, there exists complicated relational structure in the data, by which the memories can be linked together into graphs to propagate information. Typical examples include tree structure of a sentence and knowledge graph in a dialogue system. In this paper, we present a novel graph enhanced memory network GEMN to integrate relational information between memories for prediction and reasoning. Our approach introduces graph attentions to model the relations, and couples them with content-based attentions via an additional neural network layer. It thus can better identify and manipulate the memories related to a given question, and provides more accurate prediction about the final response. We demonstrate the effectiveness of the proposed approach with aspect based sentiment classification. The empirical analysis on real data shows the advantages of incorporating relational dependencies into the memory networks.

1 Introduction

Memory network [45, 39, 12, 23] has recently attracted increasing attention due to its success in many applications, such as machine reading and understanding, visual and textual question answering [2, 46, 15, 47, 42, 13]. In general, a memory network embeds a set of facts and knowledge in vector spaces as memory cells (shorten as memories). Given a question (typically represented with natural language), the model searches the supporting memories, and infers the final answer via manipulating the retrieved memories based on attention mechanism [1, 26]. The major advantage of memory networks is that they introduce an external memory component and the associated computational modules in the neural network framework to explicitly store, update, access, and manipulate the knowledge and facts for prediction, inference and reasoning given the questions. The reader and writer functions of memory networks are fully differentiable such that the entire architecture can be learned end-to-end with backpropagation.

Most of the recent works on memory networks mainly focus on the contents of the facts and knowledge. However the relations between them are not taken into account. In many cases, the facts and knowledge are not independent of each other, but are linked into a relational structure. The information exists not only in the content of the facts, but also in the relations between them. The importance of relational information has been demonstrated in the literature, see e.g., probabilistic models [7, 10, 29] and neural network models [3, 6, 21, 37].

In this paper, we propose a novel graph enhanced memory network (GEMN) to integrate the relational information into (deep) memory networks. GEMN allows for information propagation between memories and can thus better identify and manipulate the related memories to predict or reason the final response to a question. In particular, we link memories into a graph with their relations, and introduce an extra attention, *graph attention*, which is a weight vector, to capture the relational information. We model the graph attention with a Gaussian random field, i.e., a Gaussian distribution having graph Laplacian as kernels [48, 49, 35]. The memories with a short distance on the graph show strong correlation and thus likely have similar importance (i.e. weights). The graph attentions are then combined with the content-based attentions with an additional neural network layer. This introduces extra flexibility to automatically learn the combination of the two types of information (content and relations) from the data. The GEMN approach can effectively identify and leverage the important memories for a given question, and thus leads to a better inference and reasoning about the final response. There are few works investigating relational information in memory networks. An recent literature on structured attentions [19] is most relevant to our work. It models probabilistic dependencies with conditional random field that mainly focuses on sequence structures, rather than relations in general. In contrast, our approach incorporates relational information into memory networks and can model both content and relations of memories in an elegant and flexible way. The relational structure modeled in our approach can be any form, such as sequences, trees or graphs. The proposed graph attentions, combined with the content-based attentions, improves the inference and reasoning of memory networks.

We apply the proposed GEMN method to aspect level sentiment classification. With the exponential growth of user-generated content on online social network services, extracting useful insights such as preferences and opinions of users is of growing interest. Sentiment analysis [25, 31, 5, 36] focuses on detecting opinions and emotions of users on products, services and social events from large collections of texts. Typically the sentiment analysis is to estimate the positive or negative polarity of a given sentence. A more important and complicated task is to extract the sentiment polarity towards aspects [34, 25, 31, 33]. For example, in a customer review on a laptop, "*price is ok, but resolution is low!*", there are positive emotion on the aspect "*price*", and negative emotion on the aspect "*resolution*". Simply classifying the sentence as positive or negative may not properly elicit user's opinions, thus a fine-grained analysis on aspect level sentiment is necessary. We consider a supervised case where the aspects are

given. There are different approaches explored in the literature, such as SVM [22, 44, 4], conditional random field [14, 43], and neural networks [8, 41, 42, 30]. Our approach exploits graph- and content-attentions to position the related words (i.e. memories) in a sentence w.r.t. a given aspect (i.e. question of interest), and estimates the aspect level sentiment polarity based on the discovered relevant words. The empirical analysis on the real data about customer reviews on laptops and restaurants [33] demonstrates the superiority of the proposed approach.

We start the rest of this paper with a brief review on memory networks, and then introduce the graph enhanced memory network with the application to aspect based sentiment classification in the section 3. Before conclusion, the empirical analysis of the GEMN approach is presented in the section 4.

2 Memory Networks

Memory Networks [45, 39, 12, 23] are a class of learning methods with a memory component that can be read and written for prediction, inference and reasoning. The memory networks typically consist of memories and four computational modules, including I (input), G (generalization), O (output), and R (response) [45]. They are defined as follows:

- Memories are an array of vectorized objects or facts;
- Input module computes the feature representation of the input information;
- Generalization module updates the old memory with the new input;
- Output module produces an output vector given the question of interest and the current memories;
- Response module generates the final response (such as a textual answer to a question) conditioned on the output.

In general, the input and the generalization modules map the facts and the question q (e.g. a question sentence for a question-answering system) into a feature space, and get the vector representation $\mathbf{m}_i$'s and $\mathbf{u}$ for the facts and the question, respectively. The output module manipulates the memories $\mathbf{m}_i$'s and the question vector $\mathbf{u}$ to generate a single output $\mathbf{o}$ for computing the final response. In the recent literature, the output module is typically based on the attention mechanism [1, 26]. In particular, the output can be computed as (see e.g. [39]):

$$p_i = softmax(f(\mathbf{m}_i, \mathbf{u})), \;\; \mathbf{o} = \sum\nolimits_i p_i \mathbf{c}_i, \tag{1}$$

Intuitively p_i specifies how important the memory $\mathbf{m}_i$ is w.r.t. the question $\mathbf{u}$, and is scaled with the softmax function, i.e. $softmax(x_i) = \exp(x_i)/\sum_j exp(x_j)$ to ensure the constraints $\sum_i p_i = 1$, $p_i \in [0, 1]$. The weight function $f(\mathbf{m}_i, \mathbf{u})$ quantifies the relevance or similarity between $\mathbf{m}_i$ and $\mathbf{u}$. There are different definitions on the weight function. The typical choices include:

$$f(\mathbf{m}_i, \mathbf{u}) = \begin{cases} \mathbf{u}^T \mathbf{m}_i & \text{dot} \\ \mathbf{u}^T A \mathbf{m}_i & \text{general} \\ \mathbf{v}^T \tanh(A[\mathbf{u}; \mathbf{m}_i]) & \text{concatenation,} \end{cases} \tag{2}$$

where the matrix A and the vector $\mathbf{v}$ are parameters to be learned with back-propagation [26]. The output $\mathbf{o}$ is a weighted sum of $\mathbf{c}_i$. $\mathbf{c}_i$ is known as output memory, i.e. the vector representation of the fact i in the output feature space. In many cases, one can use the same embedding function and get $\mathbf{c}_i \equiv \mathbf{m}_i$ [39]. Given the output $\mathbf{o}$, the final answer to the question $\mathbf{u}$ is modeled as a classification problem. The probability of the label is predicted with softmax

$$p(s) = softmax(g(\mathbf{o}, \mathbf{u})), \tag{3}$$

where the function g can be similarly defined as (2).

3 Graph Enhanced Memory Networks

Memory networks provide a sophisticated neural network architecture to jointly model the facts for answering the questions of interest in an end-to-end fashion. However most existing methods in the literature mainly consider the content of the facts without the relations between them (such as the sentence tree structures and the knowledge graphs). In this paper, we propose a graph enhanced memory network (GEMN), which introduces additional graph attentions to model the relational information for better positioning and manipulating the relevant memories w.r.t. the given questions.

Attentions can be viewed as an additional hidden layer in a neural network framework to estimate a categorical distribution $(p_1, \ldots, p_N)$ for soft selection over the number N of memories. It is obvious that integrating the relational information into the learning process can lead to a more accurate attention distribution. Inspired by [48, 49, 35], we introduce an auxiliary random variable z_i to each memory. The value of z_i specifies graph attention weight, i.e., to what extent the memory i contributes to the output $\mathbf{o}$ based on its relations to other memories. z_i's are not independent of each other, but are interconnected into a weighted graph $\mathcal{G} = (Z, E, W)$, where $z_i \in Z$ (one for each memory) is the vertex of the graph, and $e \in E$ is the edges between z_i's. The graph $\mathcal{G}$ is represented as an adjacency matrix W of size $N \times N$, where N denotes the number of memories. Each entry $W_{i,j}$ represents the weight of an edge $e_{i,j}$ between the memories i and j. Intuitively, the larger the weight, the stronger the correlation between the two memories, and thus the more likely the memories are assigned similar graph attentions for the output. We formulate the weight as a function of the distance $d_{i,j}$ between i and j on the graph $\mathcal{G}$. The function can be of any form, but non-negative and monotonically decreasing. It can be defined as, e.g.,:

$$\text{Squared exponential:} \quad \exp(-d^2/2\ell^2) \tag{4}$$

$$\text{Rational quadratic:} \quad (1 + d^2/2\alpha\ell^2)^{-\alpha} \tag{5}$$

$$\gamma\text{-exponential:} \quad \exp(-(d/\ell)^\gamma),\ 0 < \gamma \leq 2 \tag{6}$$

With the adjacency matrix, we now model the distribution of z_i's for a soft selection over memories. The distribution is modeled as Gaussian random field

[50, 49, 48]. In particular, the state of z_i is only conditioned on the connected random variables, and follows a Gaussian distribution. The energy, i.e. sum of clique potentials of the Gaussian random field, is thus defined as [49]:

$$E(\mathbf{z}) = \frac{1}{4}\sum_{i,j} W_{i,j}(z_i - z_j)^2. \tag{7}$$

Therefore, the distribution of z_i's is

$$\begin{aligned} p(\mathbf{z}) &\propto \exp\left(-E(\mathbf{z})\right), \\ &= \exp\left(-\frac{1}{2}\mathbf{z}^T \Delta \mathbf{z}\right), \end{aligned} \tag{8}$$

which is a Gaussian with mean zero and convariance Δ^{-1}. Δ denotes combinatorial graph Laplacian: $\Delta = D - W$, where D is a diagonal degree matrix with $D_{i,i} = \sum_j W_{i,j}$. Putting everything together, we now have the graph based output $\mathbf{o}_g$:

$$\mathbf{o}_g = \sum\nolimits_i z_i \mathbf{m}_i,\ \mathbf{z} \sim \mathcal{N}(\mathbf{0}, \Delta^{-1}). \tag{9}$$

The content based output $\mathbf{o}_c$ is computed as usual, see (1). To learn from both content and relational information, we mix the two types of outputs with different ways, e.g.:

$$\mathbf{o} = h(\mathbf{o}_c, \mathbf{o}_g) = \begin{cases} \mathbf{o}_c + \mathbf{o}_g & \text{addition} \\ \mathbf{o}_c \otimes \mathbf{o}_g & \text{multiplication} \\ B[\mathbf{o}_c; \mathbf{o}_g] & \text{concatenation} \end{cases} \tag{10}$$

Here multiplication is defined as:

$$\mathbf{o}_c \otimes \mathbf{o}_g = \sum\nolimits_i a_i \mathbf{m}_i,\ a_i = softmax(z_i \cdot f(\mathbf{m}_i, \mathbf{u})) \tag{11}$$

The parameter matrix B in concatenation makes a linear transformation from the concatenation space to the memory space, and will be learned from the data with backpropagation. Addition is actually a special case of concatenation (i.e. a special weight matrix B). Compared with addition, concatenation can provide more flexibility in learning complex combination of content and graph information from the data (e.g. different weights on dimensions).

We also extend our model to a multiple level version. The structure of the deep network is stacked as follows:

$$\mathbf{u}^{(t)} = A\mathbf{u}^{(t-1)} + \mathbf{o}^{(t-1)},\ \ p_i^{(t)} = softmax(f(\mathbf{m}_i, \mathbf{u}^{(t)})), \tag{12}$$

$$\mathbf{o}_c^{(t)} = \sum\nolimits_i p_i^{(t)} \mathbf{m}_i, \qquad \mathbf{o}^{(t)} = h(\mathbf{o}_c^{(t)}, \mathbf{o}_g), \tag{13}$$

where the stacking strategy of memory embeddings $\{\mathbf{m}_1, \ldots, \mathbf{m}_N\}$ is RNN-like, i.e., keeping the memories the same across layers [39]. At the top of the network, the final response is computed with softmax: $p(s) = softmax(g(\mathbf{o}^{(t)}, \mathbf{u}^{(t)}))$.

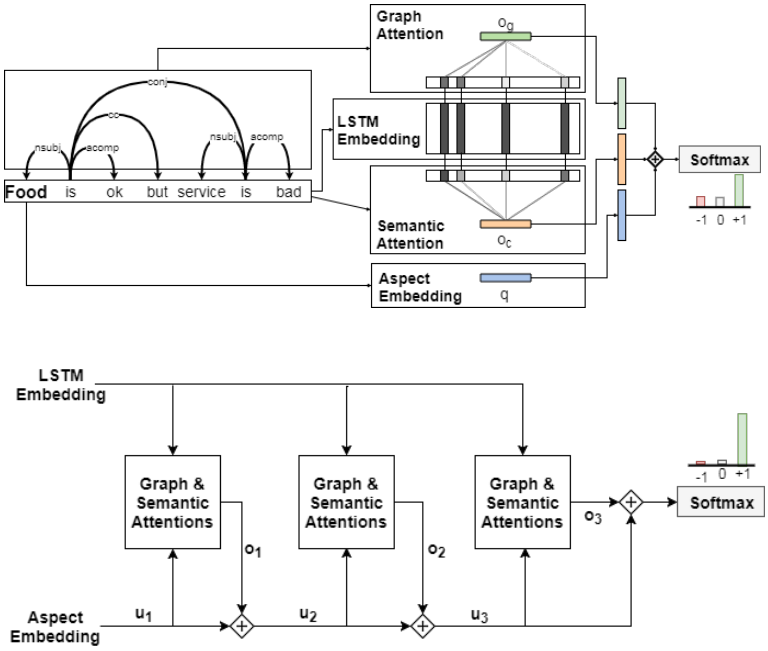


Fig. 1: Graph enhanced memory network for aspect-based sentiment classification: a single layer version (top) and a multiple layers version (bottom).

3.1 The GEMN for Sentiment Analysis

We now illustrate the graph enhanced memory network with aspect based sentiment classification. The network structure is shown as Fig. 1. Assume that there is a sentence consisting of a sequence of words $\{w_1, \ldots, w_N\}$ and multiple aspects $\{a_1, \ldots, a_M\}$. For instance, let consider a guest comment on a restaurant, "*food is ok, but service is bad*", with two aspect words "*food*" and "*service*". The task is to detect aspect level sentiment (i.e., positive emotion on "*food*" and negative emotion on "*service*") by exploiting the semantic meanings of words and the tree structure of the sentence. Here we assume each aspect only involves a single word in the sentence (e.g., "*food*" and "*service*"). In the case of multi-word aspects, the computation will be similar.

Let start with the single level version of our model, shown as the top panel of Fig. 1. In the memory network framework, the words $\{w_1, \ldots, w_N\}$ of the sentence are the facts, and the aspect word (e.g. "*food*") is formulated as the question q. The final response is the aspect level polarity (positive, negative or neutral) of the sentence. For the input module, we use word embedding [28, 32, 24] and long short-term memory (LSTM) [16, 9, 11, 40], i.e., the LSTM with pre-trained word vectors as the freezed embedding matrix. The output vector of the LSTM cell, one for each word w_i, is the memory $\mathbf{m}_i$. The question q (aspect word) is mapped as a vector $\mathbf{u}$ with word embedding. For the output module, the output

Table 1: Statistics of the datasets

Dataset	Positive	Negative	Neutral
Laptop Train	987	866	460
Laptop Test	341	128	169
Restaurant Train	2164	805	633
Restaurant Test	728	196	196

vector $\mathbf{o}$ consists of two components: content-based $\mathbf{o}_c$ and graph-based $\mathbf{o}_g$. They are computed with (1) and (9), and mixed with (10). Here the activation function for computing p_i can be flexible, e.g., we can replace the softmax with the tanh function, which is theoretically more reasonable (refer to categorical distributions of multi-label classification problem). The graph attention weights z_i's follow a Gaussian distribution (8). Since the aspect word is given, we can compute the maximum likelihood estimations (i.e. mean of the Gaussian conditioned on the aspect word) as the values of z_i's. To characterize the properties of z_i's explicitly in terms of matrix operations, the distribution (8) is expanded as:

$$\begin{bmatrix} z_a \\ \mathbf{z}_m \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} D_{a,a} - W_{a,a} & -W_{a,m} \\ -W_{m,a} & D_{m,m} - W_{m,m} \end{bmatrix}^{-1}\right) \tag{14}$$

where z_a denotes the graph attention weight of the aspect word, which is known as $z_a \equiv 1$, since the word is directly related to the aspect. The vector $\mathbf{z}_m$ denotes the unknown graph attention weights of the other words in the sentence. The Laplacian Δ is split into four corresponding blocks for the aspect word and the other words. Then the maximum likelihood estimation of $\mathbf{z}_m$ conditioned on the attention weight z_a is:

$$\mathbf{z}_m = (D_{m,m} - W_{m,m})^{-1} W_{m,a} z_a. \tag{15}$$

Finally the response module computes the final response with the softmax (3) to predict the probability of the aspect level sentiment polarity. We also model the sentiment classification problem with a multiple level version of the GEMN. The network structure is shown as the bottom panel of Fig. 1.

4 Experiments

To evaluate the performance of the graph enhanced memory network, we apply the approach to address the aspect-based sentiment classification problem. The experimental analysis is performed on real data with comparison against the state-of-the-art methods.

4.1 Datasets

The data is from the Task 4.2 of SemEval2014 [33], which includes two domain-specific English datasets for laptop and restaurant customer reviews. Each dataset

Table 2: Classification accuracy of different methods

Baselines	Laptops	Restaurants	GEMN	Laptops	Restaurants
Majority	53.45	65.00	Semantic Attention	70.69	78.84
Feature+SVM	72.10	80.89	Graph Attention	73.51	80.36
LSTM	66.45	74.28	Graph + Semantic (1 hop)	73.82	80.00
TDLSTM	68.13	75.63	Graph + Semantic (2 hops)	**74.29**	80.71
TDLSTM+ATT	66.24	74.31	Graph + Semantic (3 hops)	73.20	80.18
MemNet(1)	67.66	76.10	Graph + Semantic (4 hops)	72.88	80.54
MemNet(3)	71.74	79.06	Graph + Semantic (5 hops)	72.72	80.80
MemNet(5)	71.89	80.14	Graph + Semantic (6 hops)	72.41	**81.43**
MemNet(7)	72.37	80.32	Graph + Semantic (7 hops)	72.72	80.09
MemNet(9)	72.21	80.95	Graph + Semantic (8 hops)	72.26	80.62

has been manually labeled with annotations at the sentence level. The statistics of the datasets are summarized in Table 1. We follow the settings as in [42] that removes the sentences with the label *conflict* due to the small size of the category. The goal of the experiment is to predict the aspect level polarity (three polarities: positive, negative and neutral) of a sentence given the labeled aspect terms. Note that one sentence can include multiple aspects. For example, given the sentence "*Great food but the service was dreadful!*" and the aspect terms {"*food*" and "*service*"}, successful predictions would be {"*food*": positive and "*service*": negative}.

4.2 Baselines

The proposed method is compared with multiple recent baselines to demonstrate its performance on aspect based sentiment prediction. The baselines include:

- *Majority*: assigns to each sentence in the test set the majority sentiment label in the training set.
- *SVM* [22]: is ranked at the 1st (Laptops) and 2nd (Restaurants) places in the SemEval2014 contest. The features used in the method are sophisticated hand-crafted, including n-gram, lexicon and parse features.
- Three LSTM based models [41]: the *LSTM* directly uses the output vector of the LSTM cell for the last word of a sentence as input of a softmax to estimate the sentiment polarity. The *TDLSTM* extends the LSTM to consider the content similarity with the aspect words. The *TDLSTM+ATT* further extends the TDLSTM with the attention mechanism.
- *MemNet* [42]: uses several layers of attentions over the word embeddings. MemNet(t) denotes that the model uses t layers of attentions.

4.3 Quantitative Analysis

We first perform quantitative analysis of the proposed method. In the experiments, the GEMN is used to predict aspect level sentiment polarity (positive,

Table 3: Classification accuracy of the GEMN approach with the constituency and the dependency tree structures of the sentences

	Laptops		Restaurants	
	Constituency	Dependency	Constituency	Dependency
Semantic Attention	70.69	73.04	78.84	78.67
Graph Attention	73.51	73.51	80.36	79.20
Graph + Semantic (1 hops)	73.82	73.51	80.00	80.09
Graph + Semantic (2 hops)	74.29	74.76	80.71	80.18
Graph + Semantic (5 hops)	72.72	74.45	80.80	81.07
Graph + Semantic (10 hops)	71.16	75.39	80.54	80.08

negative and neutral) for each test sentence. The performance is measured with classification accuracy.

The graph structure of a sentence, used in the proposed approach, is extracted with Stanford's CoreNLP Toolkit [27]. Here we use the constituency tree of a sentence. The adjacency matrix is computed using squared exponential kernel with $\ell = 0.1$. The distance $d_{i,j}$ between two words is defined as the number of edges of the shortest path connecting them. The distance is normalized by the diameter of the sentence tree. The questions (i.e. the aspects) and the words are mapped as 300-dimensional Glove vectors [32], and the weights of the embedding matrix are freezed during training. The LSTM is then used to map each word in a sentence into a 128-dimensional memory space. We use an aggressive dropout of 0.7 before the final softmax layer to prevent the model from overfitting [38]. Dropout of 0.5 and 0.3 are respectively used at the input nodes and the recurrent connections of the LSTM cells. The optimization is done with Adam method [20]. The learning rate is set to 0.005. The model learns during 10 epochs with a batch size of 32 training sentences.

To get detailed performance of the proposed approach, we consider different ways to compute the output vectors for the final softmax layer:

- Variant 1: only models content based output, $\mathbf{o} \equiv \mathbf{o}_c$. In this case we do not use any information extracted from the graph structure of the sentence.
- Variant 2: only models graph based output, $\mathbf{o} \equiv \mathbf{o}_g$. Here the content based information (i.e. the semantic meanings of the words) is ignored.
- Variant 3: combines both outputs with multiplication, $\mathbf{o} = \mathbf{o}_c \otimes \mathbf{o}_g$.
- Variant 4: models the stacked and combined outputs $\mathbf{o}^{(t)}$ with (12) (t hops).

The experimental results are summarized in Table 2. For a fair comparison, we directly use the results of the baselines reported in [42]. Our approach, which models both graph and content attentions and refines over multiple layers, outperforms the baselines. It is interesting to note that our approach with only the graph attentions performs rather well. It reveals that the relational structures are pretty informative in predicting the relevant memories in the given context. The semantic information of the words, which may not be fully contained in the parse trees, can further improve the predictions. Therefore, combining both

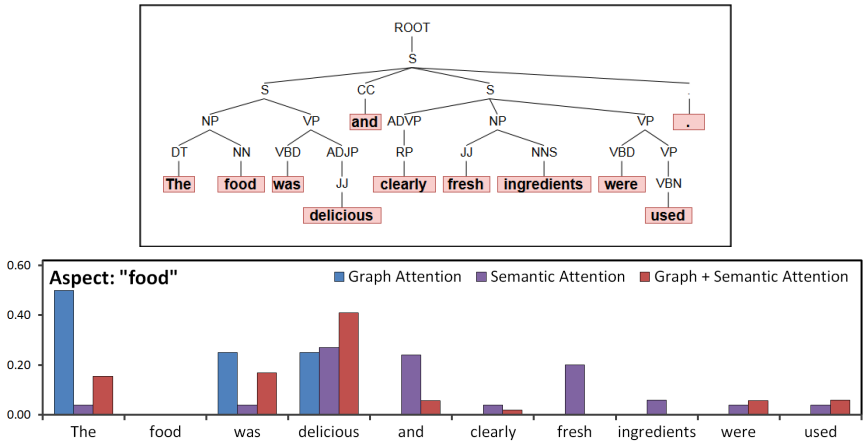


Fig. 2: Example sentence with a single aspect *food*: the constituency tree of the sentence (top) and the learned attention weights (bottom).

graph and semantic attentions leads to a notable gain in prediction accuracy. Stacking several layers to get a deep network performs well, e.g., a 0.47% increase in accuracy on the Laptops dataset. In summary, the empirical results demonstrate that, as the relational information reveals additional correlations among memories, the proposed graph attentions help the memory model to focus on the important memories w.r.t. the given aspects, and thus the GEMN achieves better predictions.

We also investigate the influence of the different graph structures of the sentences on the performance of the proposed method. Here we consider two types of tree structures: constituency and dependency. Dependency tree models one-to-one correspondence, and focuses on word grammars, while constituency tree models one-to-one-or-more correspondence, and focuses on phrase structure grammars. The dependency trees are smaller than the constituency ones. The differences between the two types of trees are investigated in details in [18]. Here we do not use the chain structure of the sentences (i.e. distance between the indices of the words), as the chain may meet difficulties in modeling some scenarios, e.g. "*service is bad, but food of the restaurant is good*". The word "*food*" should be related to "*good*", rather than "*bad*", although the index distance is larger. The tree structures of the sentences can better encode such complicated cases. In the experiments, we parse the sentences with Stanford's CoreNLP Toolkit [27] to get the constituency trees, and with the spaCy Toolkit [17] to get the dependency ones. As summarized in Table 3, the GEMN approach with the different tree structures of the sentences achieves similar performance. One can find that both types of tree structures can be well integrated with graph attentions, and provide improvement of classification accuracy. The experimental results further demonstrate the advantages of the proposed approach.

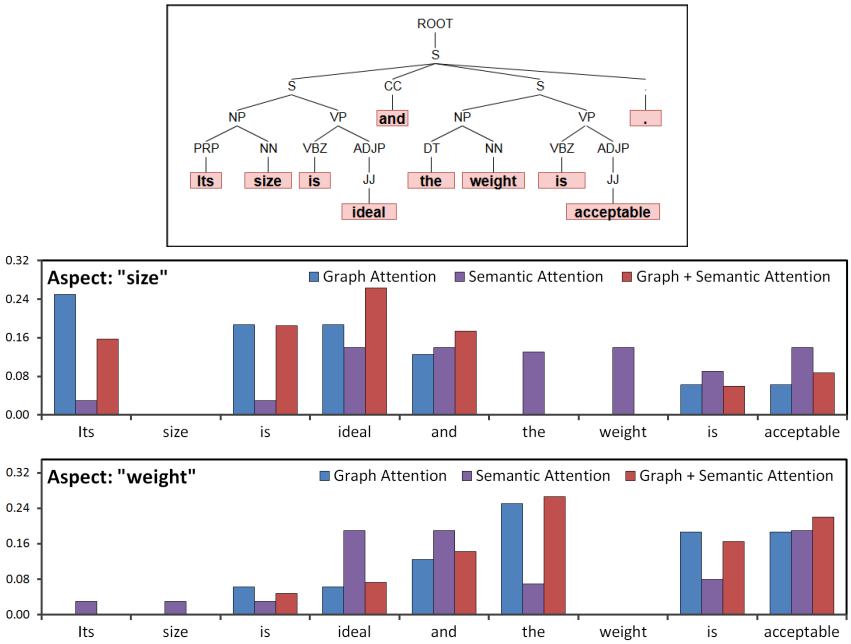


Fig. 3: Example sentence with two aspects: the constituency tree of the sentence (top), and the learned attention weights for the aspect *size* (middel) as well as the aspect *weight* (bottom).

4.4 Qualitative Analysis

To better understand the performance of the proposed approach, we further analyse the computed attentions and reveal interesting insights. Figure 2 and Figure 3 show two example sentences with one and two aspects, respectively. One can see the constituency trees of the sentences and the learned attention weights with respect to the corresponding aspect word. On one hand, graph attention, which only models the relations between memories (i.e. sentence structure), appears to effectively identify the important memories related to the context (aspect), and assigns them high weights. On the other hand, content-based (i.e. semantic) attention only considers the meanings of the words without syntactic clues. As a consequence, it highlights all the sentiment keywords, even if it is not related to the context. For example, "*fresh*" is actually not for the aspect word "*food*" in Fig. 2, and "*ideal*" not for the aspect word "*weight*" in Fig. 3. The mixed attention (graph + semantic) takes advantage of both relational and content information to identify in-context memories, and thus better discovers the important words w.r.t. the given aspects. One can see that the words "*delicious*" (Fig. 2 for the aspect "*food*"), "*ideal*" (Fig. 3 for the aspect "*size*") and "*acceptable*" (Fig. 3 for the aspect "*weight*") capture larger attentions with scores of 0.41, 0.26 and 0.22 respectively.

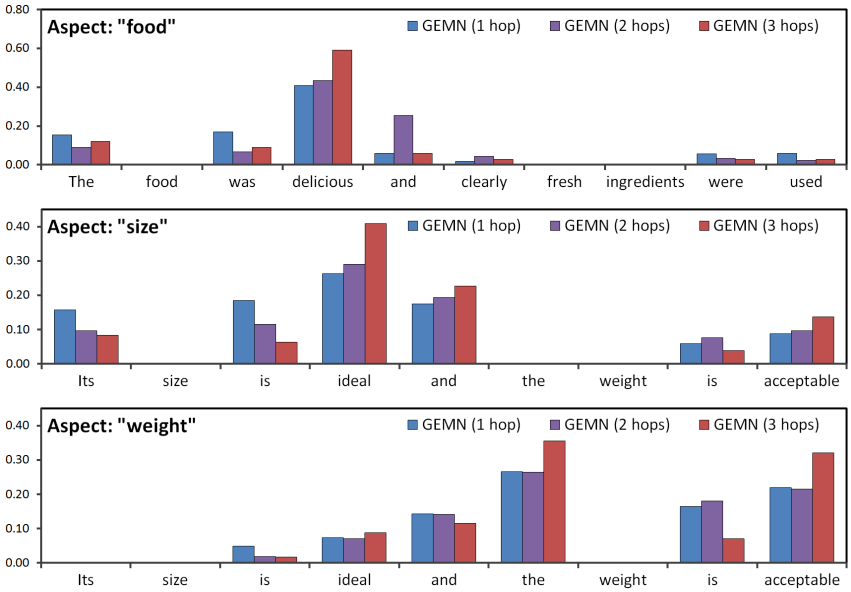
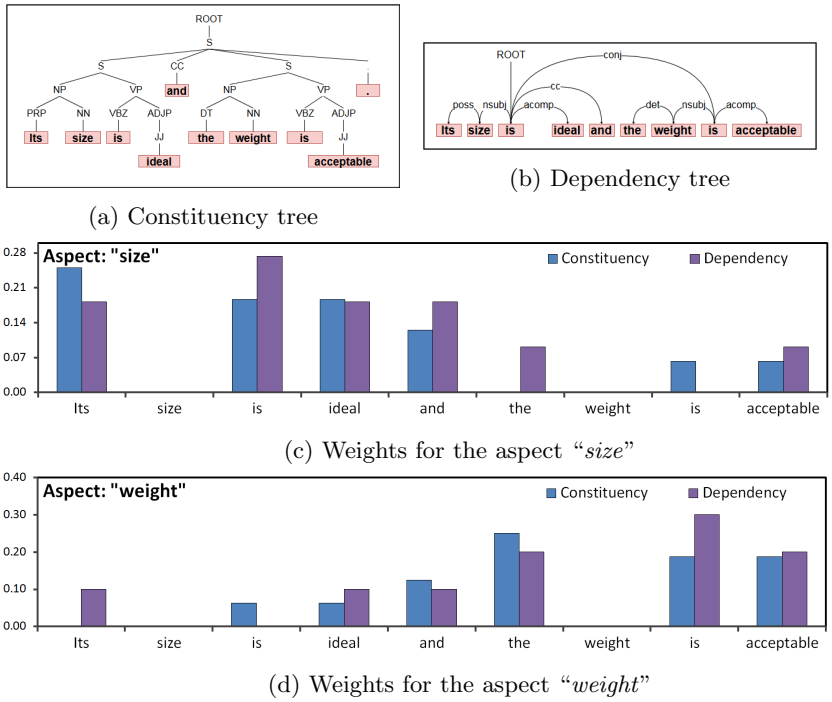


Fig. 4: Examples of the learned attention weights at each hop.

We also perform analysis on the effectiveness of a deep structure (i.e., multiple layers attentions). As shown in Fig. 4, one can find that one layer of mixed attention is not always enough to handle complex sentences. With more layers, the attention weights appear to be refined at each pass and gradually focus on the important words. For example, the word "*delicious*" has its score increasing from 0.41 to 0.59 with the number of layers, and simultaneously the noise caused by other words in the sentence is reduced. In addition, we investigate the influence of the different tree structures of the sentences on the graph attentions. Fig. 5 illustrates with an example sentence with two aspects. Although the exact values of the attention weights learned from the two tree structures are slightly different, they reveal similar tendency: the words having close syntactic relations with the aspect words get larger attention weights and vice versa.

5 Conclusion

In this paper we present a graph enhanced memory network (GEMN) to incorporate relational information for better predicting and reasoning the final response to the question of interest. We introduce a new type of attentions, graph attentions, to model the graph structure of the memories. The graph attentions are mixed with the content based attentions via an additional neural network layer, which flexibly learns the combination of both relational and content information. In turn the GEMN can better identify and manipulate the relevant memories w.r.t. the given question. The GEMN is applied to aspect-based sen-



(a) Constituency tree

(b) Dependency tree

(c) Weights for the aspect "*size*"

(d) Weights for the aspect "*weight*"

Fig. 5: Graph attentions learned from the different tree structures

timent classification, and the empirical analysis on real data demonstrates superior performance. Our work provides interesting avenues for future work, such as graph enhanced memory networks for question-answering and knowledge graph reasoning.

References

1. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. In: Proceedings of the International Conference on Learning Representations (2015)
2. Bordes, A., Usunier, N., Chopra, S., Weston, J.: Large-scale simple question answering with memory networks. ArXiv preprint:1506.02075 (2015)
3. Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., Yakhnenko, O.: Translating embeddings for modeling multi-relational data. In: Advances in Neural Information Processing Systems 26. pp. 2787–2795 (2013)
4. Brychcin, T., Konkol, M., Steinberger, J.: Uwb: Machine learning approach to aspect-based sentiment analysis. In: Proceedings of the 8th International Workshop on Semantic Evaluation. pp. 817–822 (2014)
5. Dai, A.M., Le, Q.V.: Semi-supervised sequence learning. In: Advances in Neural Information Processing Systems (2015)

6. Dai, H., Dai, B., Song, L.: Discriminative embeddings of latent variable models for structured data. In: ICML (2016)
7. De Raedt, L., Kersting, K., Natarajan, S., Poole, D.: Statistical Relational Artificial Intelligence: Logic, Probability, and Computation. Synthesis Lectures on Artificial Intelligence and Machine Learning, Morgan & Claypool Publishers (2016)
8. Dong, L., Wei, F., Tan, C., Tang, D., et al.: Adaptive recursive neural network for target-dependent twitter sentiment classification. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (2014)
9. Gers, F.A., Schmidhuber, J., Cummins, F.: Learning to forget: Continual prediction with lstm. Neural Computation 12(10), 2451–2471 (2000)
10. Getoor, L., Taskar, B. (eds.): Introduction to Statistical Relational Learning. MIT Press (2007)
11. Graves, A.: Supervised Sequence Labelling with Recurrent Neural Networks. Studies in Computational Intelligence, Springer (2012)
12. Graves, A., Wayne, G., Danihelka, I.: Neural turing machines. ArXiv preprint:1410.5401 (2014)
13. Graves, A., Wayne, G., Reynolds, M., Harley, T., Danihelka, I., et al.: Hybrid computing using a neural network with dynamic external memory. Nature 538, 471–476 (2016)
14. Hamdan, H., Bellot, P., Bechet, F.: Lsislif: Crf and logistic regression for opinion target extraction and sentiment polarity analysis. In: Proceedings of the 9th International Workshop on Semantic Evaluation. pp. 719–724 (2015)
15. Hill, F., Bordes, A., Chopra, S., Weston, J.: The goldilocks principle: Reading children's books with explicit memory representations. ArXiv preprint:1511.02301 (2015)
16. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation 9(8), 1735–1780 (1997)
17. Honnibal, M., Johnson, M.: An improved non-monotonic transition system for dependency parsing. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. pp. 1373–1378 (2015)
18. Hudson, R.: Constituency and depdency. Linguistics 18, 179–198 (1980)
19. Kim, Y., Denton, C., Hoang, L., Rush, A.M.: Structured attention networks. In: Proceedings of the International Conference on Learning Representations (2017)
20. Kingma, D., Ba, J.: Adam: A method for stochastic optimisation. In: Proceedings of the International Conference on Learning Representations (2015)
21. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. In: Proceedings of ICLR (2017)
22. Kiritchenko, S., Zhu, X., Cherry, C., Mohammad, S.: Nrc-canada-2014: Detecting aspects and sentiment in customer reviews. In: Proceedings of the 8th International Workshop on Semantic Evaluation. pp. 437–442 (2014)
23. Kumar, A., Irsoy, O., Ondruska, P., Iyyer, M., Bradbury, J., Gulrajani, I., Zhong, V., Paulus, R., Socher, R.: Ask me anything: Dynamic memory networks for natural language processing. In: Proceedings of the International Conference on Machine Learning (2016)
24. Le, Q.V., Mikolov, T.: Distributed representations of sentences and documents. In: Proceedings of the 31th International Conference on Machine Learning. pp. 1188–1196 (2014)
25. Liu, B.: Sentiment Analysis and Opinion Mining. Morgan & Claypool Publishers (2012)

26. Luong, M.T., Pham, H., Manning, C.D.: Effective approaches to attentionbased neural machine translation. In: Proceedings of the Conference on Empirical Methods in NLP (2015)
27. Manning, C.D., Surdeanu, M., Bauer, J., Finkel, J., et al.: The Stanford CoreNLP natural language processing toolkit. In: Association for Computational Linguistics (ACL) System Demonstrations. pp. 55–60 (2014)
28. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in Neural Information Processing Systems 26. pp. 3111–3119 (2013)
29. Miller, K.T., Griffiths, T.L., Jordan, M.I.: Nonparametric latent feature models for link prediction. In: Advances in Neural Information Processing Systems (2009)
30. Nguyen, T.H., Shirai, K.: Phrasernn: Phrase recursive neural network for aspect-based sentiment analysis. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. pp. 2509–2514 (2015)
31. Pang, B., Lee, L.: Opinion mining and sentiment analysis. Foundations and Trends in Information Retrieval 2(1-2), 1–135 (2008)
32. Pennington, J., Socher, R., Manning, C.D.: Glove: Global vectors for word representation. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing. pp. 1532–1543 (2014)
33. Pontiki, M., Galanis, D., Pavlopoulos, J., Papageorgiou, H., et al.: Semeval-2014 task 4: Aspect based sentiment analysis. In: Proceedings of the 8th International Workshop on Semantic Evaluation. pp. 27–35 (2014)
34. Schouten, K., Frasincar, F.: Survey on aspect-level sentiment analysis. IEEE Trans. Knowledge and Data Engineering 28(3), 813–830 (2016)
35. Smola, A.J., Kondor, R.: Kernels and regularization on graphs. In: Proceedings of the conference on Learning Theory (2003)
36. Socher, R., Perelygin, A., Wu, J.Y., Chuang, J., et al.: Recursive deep models for semantic compositionality over a sentiment treebank. In: EMNLP (2013)
37. Socher, R., Chen, D., Manning, C., Ng, A.Y.: Reasoning with neural tensor networks for knowledge base completion. In: Advances in Neural Information Processing Systems. pp. 926–934 (2013)
38. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: A simple way to prevent neural networks from overfitting. Journal of Machine Learning Research 15(1), 1929–1958 (2014)
39. Sukhbaatar, S., Szlam, A., Weston, J., Fergus, R.: End-to-end memory networks. In: NIPS. pp. 2431–2439 (2015)
40. Sutskever, I.: Training Recurrent Neural Networks. Ph.D. thesis, University of Toronto (2013)
41. Tang, D., Qin, B., Feng, X., Liu, T.: Target-dependent sentiment classification with long short term memory. ArXiv preprint:1512.01100 (2015)
42. Tang, D., Qin, B., Liu, T.: Aspect level sentiment classification with deep memory network. In: EMNLP. pp. 214–224 (2016)
43. Toh, Z., Wang, W.: Dlirec: Aspect term extraction and term polarity classification system. In: Proceedings of the 8th International Workshop on Semantic Evaluation. pp. 235–240 (2014)
44. Wagner, J., Arora, P., Cortes, S., et al.: Dcu: Aspect-based polarity classification for semeval task 4. In: Proceedings of the 8th International Workshop on Semantic Evaluation. pp. 223–229 (2014)
45. Weston, J., Chopra, S., Bordes, A.: Memory networks. In: ICLR (2015)
46. Xiong, C., Merity, S., Socher, R.: Dynamic memory networks for visual and textual question answering. In: ICML (2016)

47. Xu, H., Saenko, K.: Ask, attend and answer: Exploring question-guided spatial attention for visual question answering. In: ECCV (2016)
48. Xu, Z., Kersting, K., Tresp, V.: Multi-relational learning with gaussian processes. In: IJCAI. pp. 1309–1314 (2009)
49. Zhu, X., Ghahramani, Z., Lafferty, J.: Semi-supervised learning using Gaussian fields and harmonic functions. In: ICML (2003)
50. Zhu, X., Lafferty, J., Ghahramani, Z.: Semi-supervised learning: From gaussian fields to gaussian processes. Tech. Rep. CMU-CS-03-175, Carnegie Mellon University (2003)